CHAPTER 2

# The HTML 4 Way

In This Chapter

The problem of browsers having different and incompatible extensions

An introduction to cascading style sheets

The HTML 4 Way simply put

The future of HTML—XML?

Let's talk for a moment about history. Not ancient history but, say, mid-1997 Web-publishing history. What were people's biggest concerns about publishing a page back then?

- **Compatibility across browsers.** Would a page that looked awesome in Internet Explorer 4 look right in Netscape Navigator 4? Would JavaScript, which animated a figure perfectly in Navigator 4, work in Internet Explorer 4? And what if a visitor to your site was using Netscape Navigator 2? Did you really have to maintain a nonframes-compatible version of every single page?
- **Keeping content fresh.** People expect to come to your Web site to get accurate, up-to-date answers to their questions. Can you afford to be constantly updating your content when it is buried in a bunch of HTML? You practically need a Web developer just to correct a misspelling, when the content and the design are so closely interwoven.
- **Keeping a current look.** The standards for gorgeous Web page design are climbing every week. Keeping your site looking hot is both costly and time-consuming.

HTML 4 can't completely solve all these problems, but it can reduce the magnitude of these problems. This chapter explains how.

## The Extension Problem

In America, we value innovation. We celebrate people who make a lot of money by being the first to implement great ideas. This culture of reverence for mavericks creates a lot of problems for Web developers when the mavericks are producing the browsers that have the lion's share of the market.

Both Netscape and Microsoft are to blame (don't try to pin it all on Bill Gates, founder of Microsoft, the world's richest man, and the subject of an entire genre of humor); both these companies have introduced proprietary *extensions* (ways of marking up HTML) that only work on their own browsers.

AOL's purchase of Netscape hasn't changed the browser race drastically, but it has added a layer of mystery to the continuing evolution of Netscape Communicator. Of course, Netscape and Microsoft aren't doing anything car manufacturers don't do every year. When the first minivan manufacturer added cupholders, did anyone complain? This feature may have made you purchase that minivan instead of the competitor's. Today, every minivan has cupholders — so it goes with great innovations!

So what do cupholders and mavericks have to do with your Web site? Everything. The people who make browsers are trying to get more people to purchase and use their own browsers — rather than the competitors — by adding more new features. Of course, you're thinking, isn't this contrary to the idea of the Web, that everything would be standard so everything could be seen on every computer, regardless of platform or software installed? You've obviously been paying attention. It is, indeed, antithetical to the very basis of the Web. In fact, if these motivations had been in place when the Web got off the ground, we'd probably have two or three different Webs and nothing from one could be viewed on either of the others!

Let's look at the specific problems browser manufacturers introduce when they add new features no one else has.

## Compatibility across browsers

When Netscape introduced frames with Version 3 of Navigator, designers had a problem: Did they start using a feature that not only couldn't be seen by a visitor who came to their sites from any other browser, but couldn't even be seen by a visitor who came to their sites from all previous versions of Netscape Navigator? The answer is many of them started to use frames, but maintained two separate sites: one for people coming to the site with Netscape Navigator 3 and one for everyone else. They also put little icons (provided by Netscape at no charge) on their pages saying: This site looks best when viewed with Netscape Navigator. What could be more of a pain for designers?

What could be better for Netscape? As it turned out, Microsoft ended up adding support for frames to their own browsers, making frames a standard. Now, both Netscape Communicator 4.7 and Internet Explorer 5.0 support frames. Most sites look identical in either Netscape or IE, as long as you're running the newest version of the browser.

So, didn't it all end happily ever after? Not exactly. This constant battle for market share that rages between Microsoft and Netscape means as a Web designer, you will constantly evaluate your material to see what works with which browser. You will always have to assess the trade-off between the following three positions:

1. Don't use any tags that aren't widely supported. The result of this decision is you will only have to support one set of pages (rather than one for the latest set of extensions and one for everyone who has browsers that don't support those). But — and this is the major drawback — your site will never look as cool as the sites of those who do use the latest extensions. You won't be able to implement the latest and greatest design features.
2. Use the proprietary tags and to heck with people who haven't upgraded. Arrogant but, for certain sites, this works. If your site is about design, you don't have any choice but to use the latest design tools. (No one attends fashion shows of last year's clothes.) If, however, you are trying to sell something on your site (other than your own design services), then you'd better be a little less cavalier about your visitors' experiences. If your site appeals to education or social causes — where you can do the most good if you reach the maximum number of people — then you should probably not adopt this attitude.
3. Support two versions of the same site: one that uses the latest extensions and one that doesn't. Later in this book, you learn how to use JavaScript to find out which browser your visitors are using when they come to your page and then to show them the appropriate page for their browsers. Of course, this approach is the most expensive, because you have to maintain the site in two versions — what a pain!

How did it happen that on this cross-platform Web, where everything was supposed to work on every computer, we ended up in a situation where we are writing browser-specific pages? Isn't someone supposed to keep this from happening?

Indeed, someone is trying.

## The World Wide Web Consortium

The World Wide Web Consortium (W3C) is the organization created to set standards for the Web. However, the W3C has no means of enforcement (and no inspection teams to make sure soon-to-be-released browsers comply). The W3C has a small staff of its own, but most of the people on the committees (about which you'll soon hear more) who write the standards come from the academy and industry. Sometimes those who sit on committees represent specific interests.

While the theory goes that the W3C sets a standard and the industry follows, the reality is, the W3C is usually following the actions taken by the largest browser manufacturers: Netscape and Microsoft. Because Microsoft and Netscape continue to add their own proprietary extensions, the W3C is often in the position of simply compiling a list of the features already in use on either browser and deciding which ones will be declared standard. The W3C also does a lot of interesting work in developing standards for ancillary Web technologies.

# The Ideal: Separating Structure from Presentation

The most expensive thing about publishing a Web site isn't the cost of your account with an ISP. It isn't the cost of any tool you may purchase to increase your productivity when writing HTML. It isn't the time spent learning HTML. What is it? It's the time spent maintaining your site.

HTML 4 tackles this problem head-on by storing all the information about the way things should look in a separate location (either in a separate place in the same document or, better yet, in a separate file altogether) from the content. You simply indicate what type of element (paragraph, title, list, heading) each block of text is, using HTML, and then, in a style sheet, you describe how you want each element to look.

## Cluttered HTML (the pre-HTML-4 universe)

Unless you are content to let your content get stale and your style become dated, you need to put time into updating both your design and your content. In the pre-HTML-4 world, you needed to hire a Web developer to do both of these, which became expensive. Why did you need a Web developer simply to update your content? Consider the following HTML:

```
<li><FONT SIZE="+1" FACE="comic sans ms" FAMILY="sans-serif"
COLOR="#0000FF"><P><A name="do"></a><B>What does <i>Stay In
Touch</i> do?</B></P></FONT>
<FONT SIZE="-1" FACE="comic sans ms" FAMILY="sans-serif"
COLOR="#000000"><P><i>Stay In Touch</i> allows you to harness
the power of the World Wide Web to communicate with people who
visit your web site. Using <i>Stay In Touch</i> list management
service you can set up a sign-in page on your web site today
```

```
and customize it to match the rest of your web site. Your
visitors can sign into your site when they visit, then you can
send mail to your visitors based on a number of criteria: the
interest they indicate, the publications they read, etc. To see
an example of this, go to the Demo and view the Send Mail
option.</P></FONT>
<li><FONT SIZE="+1" FACE="comic sans ms" FAMILY="sans-serif"
COLOR="#0000FF"><P><A name="security"></a><B>How secure is my
list?</B></P></FONT>
<FONT SIZE="-1" FACE="comic sans ms" FAMILY="Sans Serif"
COLOR="#000000"><P>Only you have access to your list. Access to
your list is available exclusively from secure pages residing
on our server. You have enough to worry about. The security of
your list needn't be one of those things.</P></FONT>
```

Figures 2-1 and 2-2 are what it looks like in your browser on a PC and then on a Mac (notice the font is slightly different).

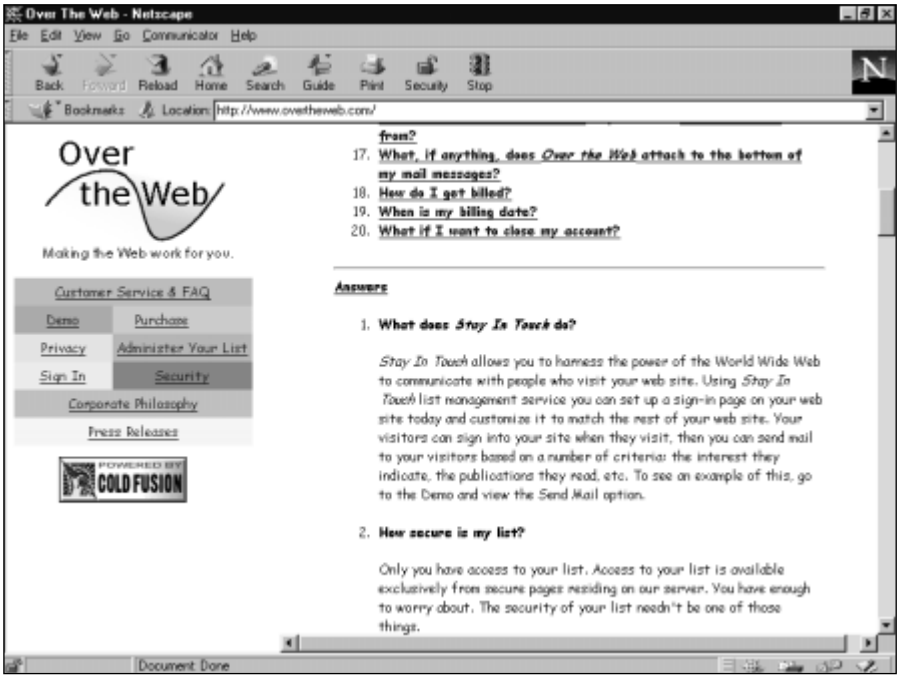
Over The Web - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Guide Print Security Stop
Bookmarks Location: http://www.overtheweb.com/
Over the Web
Making the Web work for you.
Customer Service & FAQ
Demo
Purchase
Privacy
Administer Your List
Sign In
Security
Corporate Philosophy
Press Releases
POWERED BY COLD FUSION
from?
17. What, if anything, does Over the Web attach to the bottom of my mail messages?
18. How do I get billed?
19. When is my billing date?
20. What if I want to close my account?
Answers
1. What does Stay In Touch do?
Stay In Touch allows you to harness the power of the World Wide Web to communicate with people who visit your web site. Using Stay In Touch list management service you can set up a sign-in page on your web site today and customize it to match the rest of your web site. Your visitors can sign into your site when they visit, then you can send mail to your visitors based on a number of criteria: the interest they indicate, the publications they read, etc. To see an example of this, go to the Demo and view the Send Mail option.
2. How secure is my list?
Only you have access to your list. Access to your list is available exclusively from secure pages residing on our server. You have enough to worry about. The security of your list needn't be one of those things.
Document Done

**Figure 2-1:** The previous text displayed in a browser on a PC

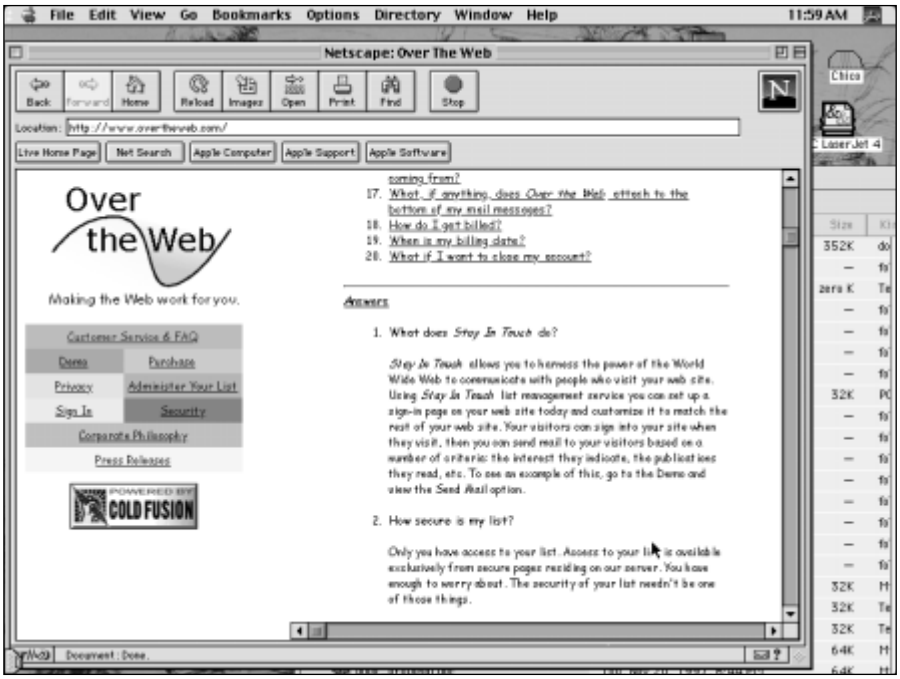
File Edit View Go Bookmarks Options Directory Window Help
11:59 AM
Netscape: Over The Web
Back Forward Home Reload Images Open Print Find Stop
Location: http://www.overtheweb.com/
Live Home Page Net Search Apple Computer Apple Support Apple Software
Over the Web
Making the Web work for you.
Customer Service & FAQ
Demo Purchase
Privacy Administer Your List
Sign In Security
Corporate Philosophy
Press Releases
POWERED BY COLD FUSION
coming from?
17. What, if anything, does Over the Web attach to the bottom of my mail messages?
18. How do I get billed?
19. When is my billing date?
20. What if I want to close my account?
Answers
1. What does Stay In Touch do?
Stay In Touch allows you to harness the power of the World Wide Web to communicate with people who visit your web site. Using Stay In Touch list management service you can set up a sign-in page on your web site today and customize it to match the rest of your web site. Your visitors can sign into your site when they visit, then you can send mail to your visitors based on a number of criteria: the interest they indicate, the publications they read, etc. To see an example of this, go to the Demo and view the Send Mail option.
2. How secure is my list?
Only you have access to your list. Access to your list is available exclusively from secure pages residing on our server. You have enough to worry about. The security of your list needn't be one of those things.
Document: Done.
Chico
LaserJet 4

**Figure 2-2:** The previous text displayed in a browser on a Mac

## The maintenance nightmare

From looking at the HTML and then seeing the HTML interpreted by the browser, you can pretty much tell what part of the text is instructions to the browser and what part is the content. But would you feel comfortable making changes to the content — say, adding another bulleted set of questions and answers? Probably not. And you probably wouldn't want someone else who didn't know what all those codes meant doing it either.

HTML 4 almost eliminates this need to have an HTML expert perform site maintenance. This means HTML 4 helps reduce the cost of maintaining your Web site! When was the last time you heard anything about reducing costs being associated with the Web?

Consider the site map in Figure 2-3. Every screen should have the same formatting: same font, same heading sizes, same alignment, same text color, and same background color. Wouldn't it be nice if you could put all this information in one place and have every page look to that page for formatting instructions?

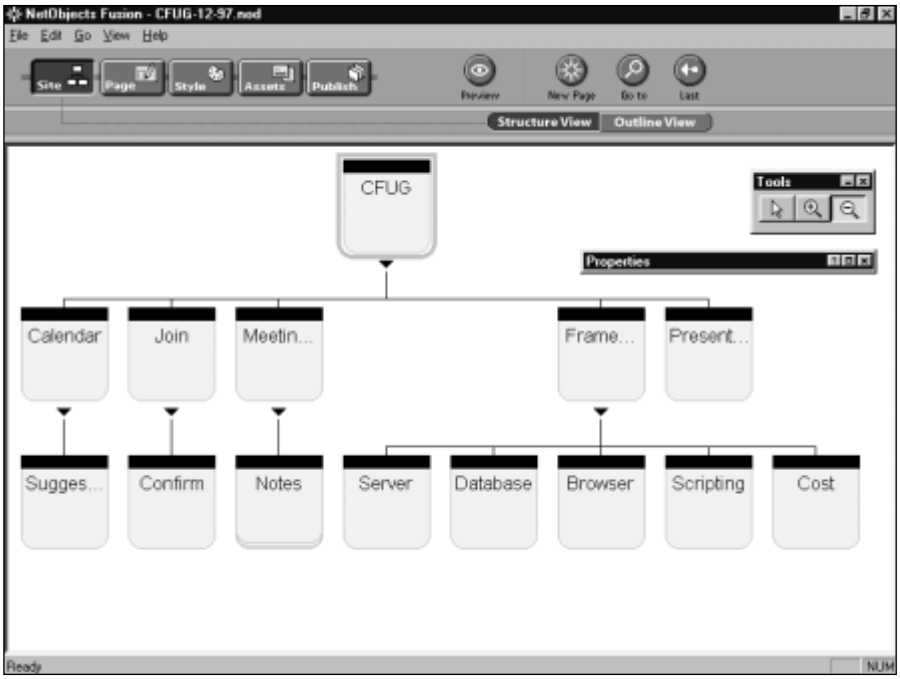


**Figure 2-3:** Site map

## HTML 4 defines structure

A brief outburst of history—Standard Generalized Markup Language (SGML), approved by the ISO in 1988, is the basis for HTML. SGML is a standard for electronic document exchange, and in fact, was used in the publication of this book. In the publication process, the book publisher marks up a text using SGML and then tells the printer to print it by providing a list of how each element of SGML should appear.

| | |
|---|---|
| **Title:** | Make this blue. 18pt. Times New Roman. Also center it. |
| **Section headings:** | Make them black (the rest of the book should all be black, by the way) and 18pt. The font should be Times Roman. They should be left-justified and triple-spaced. |
| **Paragraphs:** | Each paragraph should be indented ¾ inch. Font should be 12pt. And Times Roman. |
| **Examples of code:** | These should be in Courier 12pt. They should be indented from each margin by ¾ inch. |

The publisher uses SGML to indicate where titles, section headings, paragraphs, and code examples (referred to collectively as *elements*) begin and end. When HTML was first created, this was how it was supposed to work. But designers weren't happy with the lack of control over formatting, so they started to use it in a way that never had been intended.

HTML 4 enables us to return to the ideal of HTML. What does this mean to you and your ability to maintain a site? If the only codes that appear in your HTML are structure codes (with the presentation codes appearing elsewhere), then no reason exists why you can't use less expensive talent to maintain your content.

Consider the following code. It produces the same results as the previous example in the browser. Notice there is no formatting. All the HTML you see is related to the structure. We discuss the formatting code in the next section. For now, just remember we put the information related to presentation into a style sheet.

```
<LI>What does <i>Stay In Touch</i> do?
<P><i>Stay In Touch</i> allows you to harness the power of the
World Wide Web to communicate with people who visit your web
site. Using <i>Stay In Touch</i> list management service you
can set up a sign-in page on your web site today and customize
it to match the rest of your web site. Your visitors can sign
into your site when they visit, then you can send mail to your
visitors based on a number of criteria: the interest they
indicate, the publications they read, etc. To see an example of
this, go to the Demo and view the Send Mail option.</P>
<LI>How secure is my list?
<P>Only you have access to your list. Access to your list is
available exclusively from secure pages residing on our server.
You have enough to worry about. The security of your list
needn't be one of those things.</P>
```

How comfortable would you be updating the previous HTML? How about if you needed to add another set of questions and answers? Already, you can see using HTML 4 makes a world of difference. The information the browser must know to format the previous text is stored separately, in a style sheet.

# Introducing Cascading Style Sheets

A *style sheet* contains information about how you want your page to look. The idea and the name come from the publishing industry, where style sheets are still used today. But why *cascading* style sheets (CSS)? Good question. Most of the time, when you create a site, you want a uniform look across the site. You probably don't want each different section of a site to use a different font or a different background color. You probably also don't want one section to have all the text left-justified (like this book) and have another section right-justified, just for the heck of it. If you are creating the site yourself, this kind of uniformity is easier to maintain, but even you may sometimes forget how you did it on another page.

The answer to uniformity is a style sheet. Define your style sheet outside any of your pages (give it a name like `MyStyleSheet.css`) and link all your pages to it. Now, any element you have described in your style sheet will look the way you want it to look (and the same) in every page where that element is used.

But what if you want the color of some text to be different on one page? Say, to indicate this is a new addition to your site or a limited-time offer? Easy. Just define the style for this element right there, in that tag. Isn't this the way pre-HTML-4 worked? Exactly. Except you will use style definitions within elements only as the exception, not as the rule.

The browser knows to look in the external (global) style sheet for style definitions about each element it encounters (see Figure 2-4). It uses those unless it finds locally defined styles, and then those locally defined styles override the globally defined styles. Finally, you can define styles for an entire page. Element-level styles override page-specific styles, which override global styles.

As confusing as this sounds, most often you only have a globally defined style sheet. Still, it's nice to know you can override the global style sheet if you like.

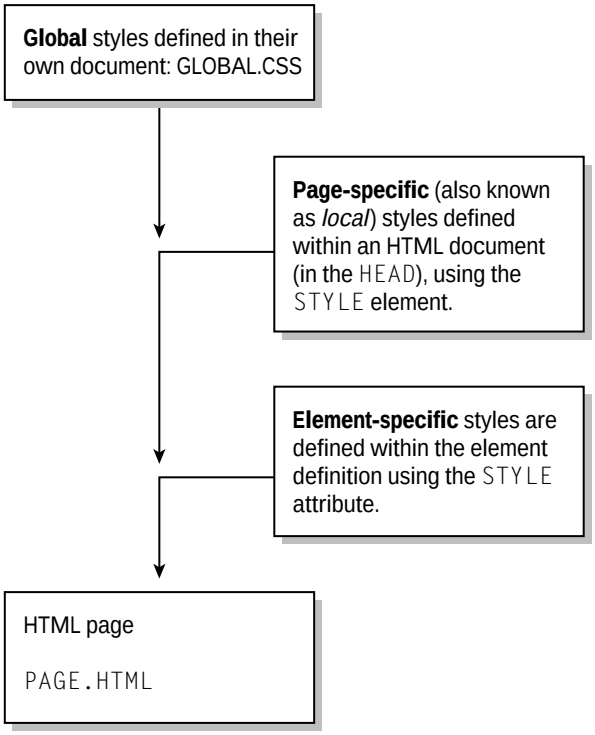
**Global** styles defined in their own document: GLOBAL.CSS
**Page-specific** (also known as *local*) styles defined within an HTML document (in the HEAD), using the STYLE element.
**Element-specific** styles are defined within the element definition using the STYLE attribute.
HTML page
PAGE.HTML

**Figure 2-4:** The cascading model of style definitions

# The HTML 4 Way

So what is the HTML 4 Way? The *HTML 4 Way* is using HTML within your document only to define structure and putting all information your browser needs about presentation into a style sheet.

The rest of this book focuses on the HTML 4 Way. If you are already a veteran Web developer, then you may have to break some bad habits. If you are new to Web development, then you have it easier; you can learn it right — right from the beginning.

# The Future: XML

Let's go back to SGML — standard generalized markup language — again. HTML came from SGML, but those folks who live and breathe SGML in publishing and the academy were not satisfied with what HTML could do. They kept complaining to the W3C about the things HTML couldn't do that SGML could.

The W3C tried to appease these SGML-zealots (using the term advisedly) who complained about the shortcomings of HTML — now remember the reason HTML became so popular so quickly was it was simple — by expanding the HTML specification. Each time a new version of HTML came out, the SGML fanatics complained at greater length (if you ever see SGML, you'll see it is not known for its brevity, unlike HTML) about its failure to perform like SGML.

You can see where this is going: XML. Extensible Markup Language (XML) was approved in January 1998 by the members of the W3C, and recommended by the W3C shortly thereafter. To help you understand why the W3C had delayed supporting the demand of the SGML advocates, note that, at the time XML was approved, neither of the major browsers supported it (please recall that every feature of HTML 4 was already supported in one of the two major browsers — and had been for almost a year — by the time it was approved). So why would you want to know or use XML? Two reasons:

- **Flexibility.** Unlike HTML, XML gives you the ability to define your own tags.
- **Power.** Unlike HTML, XML enables the use of variables you can define externally to your page. This makes XML more like a programming language. Before XML, you had to use server-side scripting, JavaScript, or Java to include variables in your documents. Because XML isn't implemented (at the time of this publication), you still must use one of these.

And why wouldn't you want to use XML?

- **Flexibility.** With flexibility comes responsibility. You have to define each of your tag definitions properly in a document type definition (DTD).

- **Complexity.** XML is a lot like a programming language. None of the programming languages have taken off the way HTML has. The average Web developer isn't ready for this kind of complexity.

We'll cover XML basics in subsequent chapters, but for the whole story, look for the *XML Bible* by Elliotte Rusty Harold (IDG Books, 1999).

# The Ideal Versus the Reality

While HTML 4 addresses all three of the major concerns of Web developers outlined as we began this chapter — compatibility across browsers, keeping content fresh, and keeping a current look — as a designer, you are still somewhat trapped by the dilemma that has frustrated all previous Web designers contending with new versions. Until all browsers support all features of HTML 4 and you can be sure all visitors to your site are using HTML-4-compatible browsers, you are left with the following options:

- **Don't use any HTML-4-specific tags or methodology.** Just keep doing things the way you have been. Clutter up your document with formatting and structural HTML and take a new job when the mess becomes too unmanageable.
- **Use HTML 4 and to heck with people who haven't upgraded to HTML-4-compatible browsers.** Use HTML for structure only within your documents. If people are viewing your site with a browser that doesn't support HTML 4, then they will see minimalist formatting, but this is basically their problem.
- **Support two versions of the same site: one that uses HTML 4 and one that doesn't.** Keep the content up-to-date in your current site. Apply any face-lifts to the HTML 4 version of your site.

# Where's All This Going?

Without question, the direction of the Web is toward the HTML 4 Way. Whether it is with HTML 4 this year or XML next year, you can count on the need for the separation of structure from design. As HTML becomes more sophisticated and more and more formatting features are added, the need for separation will become even clearer.

Web designers want to make their pages more visually compelling, taking the best practices from print and the best practices from television. Management wants the ever-spiraling cost of maintaining a Web site to level off or drop — why should they have to pay high-priced Web designers just to update the text on a page? HTML 4 meets both these needs in a concrete way.

Unlike previous versions of HTML, HTML 4 is pretty well implemented in both Navigator and Internet Explorer. With HTML 4, you'll find you can do things you only dreamed of before now. Indenting a paragraph? No problem. Indenting every paragraph? No problem. Change the text color of every heading? No problem.

The die is cast, and even though you may have to change the way you think about creating and maintaining a Web site, it is for the better. Never before have you had the kind of control over formatting you have now. Never before have you had the flexibility to make your text do whatever you want, including moving!

# What Should You Do Now?

Get excited! The face of the Web will never look the same. You are going to be part of the story, whether you are an example in the next version of this book of the bad, old way people used to write HTML or an award-winning Web designer, implementing HTML 4 to bring traffic and recognition to your Web site.

Don't necessarily look to the Web for the future of design. Few places exist where you can see the power of HTML 4 in action. Look to print — where few formatting constraints exist (although they, unlike we, are constrained by two dimensions). Look to television where animated text is something we take for granted.

# From Here

If you are an old hand at HTML and you have been bearing with us to see what this HTML 4 Way is all about, you can jump right into Part IV to learn about enhancing your presentation with Cascading Style Sheets.

If you are just getting started with HTML, proceed to Chapter 3 to learn the basics of creating your first Web page.

# Summary

The HTML 4 Way is, indeed, the future of Web publishing. It reduces the cost of maintaining a Web site, increases the speed of giving your site a periodic face-lift, and completely changes the paradigm of Web development by separating presentation from structure.

XML may or may not take over as the king of the Web for reasons, principally, of complexity. In any case, the design choices available to you as a Web designer continue to increase. Even if the W3C didn't have this as a goal, Netscape and Microsoft continue to try to sell their own browsers, which necessitates the inclusion of new features.

While it is true you can't assume every visitor to your site has the most recent browser supporting HTML 4, you can still provide adequate content to your foot-dragging visitors and stunning design to your visitors who are keeping up.

✦ ✦ ✦